ZMK
AF

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application of Laurence B. Boucher, et al.		Serial No:	10/724,588
Filing Date:	November 28, 2003	Examiner:	Wen Tai Lin
Atty. Docket No:	ALA-025	GAU:	2154

For: INTELLIGENT NETWORK INTERFACE SYSTEM AND METHOD FOR ACCELERATED PROTOCOL PROCESSING

December 5, 2006

MS Appeal Brief
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REPLY BRIEF

This is a Reply to an Examiner's Answer dated October 5, 2006. An Appeal Brief was mailed by appellants on August 2, 2006.

Argument

Appellants will not, in this Reply, reiterate the arguments made in the Appeal Brief, but will instead respond to the new arguments made by the Examiner in his Answer. As will be seen, the desperation of those new arguments actually serves to reinforce, rather than diminish, the strength of appellants' earlier arguments.

For example, the Examiner's Answer begins those new arguments by rephrasing the Grounds of Rejection of claim 1 to construe a "TCP connection" as "a physical connection established for transmission of a TCP packet."[1] Such a construction conflates the physical layer (layer 1) of network communication protocols with the transport layer (layer 4), in which TCP resides. Moreover, those of ordinary skill in the art know that

[1] Examiner's Answer, page 4, lines 2-4.

networks do not function like the old-fashioned telephone system, where a fixed physical connection was established for each phone call. Instead, networks may employ a multitude of routers that each save and redirect a packet so that a contiguous electrical connection between endpoints may never exist, even though a contiguous physical connection between endpoints always exists. Appellants respectfully submit that for these reasons, as well as others discussed below, the Examiner's construction is clearly unreasonable.

The Examiner's Answer then proceeds further out on a limb by stating: "Since TCP process 91 of Fig. 2B is designed for constructing a TCP packet, the execution of process 91 is in a way 'running protocol processing instructions.'"[2] That is, the Examiner's Answer alleges that "constructing a TCP packet" by Bennett equates to "running protocol instructions in the local host to create a TCP connection between the local host and the remote host," as recited by claim 1. If that were true, construction of the packet would somehow cause distal network routers to physically contact each other to form a new "physical connection". Moreover, every time a new packet was constructed a new TCP connection would be created, even for successive packets constructed from contiguous application data. Such an interpretation is untenable. Furthermore, such an interpretation contradicts the Examiner's later allegation that Bennett's automatic ACK generation equates to controlling the TCP connection by Bennett's "network card 2000", because there is no indication that the ACK corresponds to the TCP connection allegedly created by constructing a packet in the "TCP process 91" of "local node 1000". Indeed, by the Examiner's own reasoning, the construction of an ACK packet would create a new TCP connection,[3] so the ACK could not correspond to the TCP connection that the Examiner alleges was created by constructing a packet in the "TCP process 91". Appellants respectfully submit that for these reasons, as well as others discussed below, the Examiner's interpretation is at best unreasonable.

---

[2] *Id*, page 4, lines 4-5.

[3] The Examiner's Answer on page 10, lines 1-4 states that an ACK packet "by itself is a TCP packet", but fails to realize that by the Examiner's logic, constructing that ACK packet by "network card 2000" would create a new TCP connection that is different than the TCP connection the Examiner alleges was created earlier by constructing a TCP packet in "TCP process 91" of "local node 1000".

The Examiner's Answer continues its new arguments by citing to an "ARP (Address Resolution Protocol) request" that is not taught in the cited passage or anywhere else in Bennett, and then alleges that the unidentified ARP request "must contain at least a source MAC address and a target IP address".[4] The Examiner's Answer then presents a new allegation of inherency, stating that "the ARP is an inherent function of an Ethernet node". Although appellants do not believe that at this stage they are allowed to present evidence to rebut this new allegation of inherency,[5] the simple realization that ARP is a network layer protocol (layer 3) that may or may not be employed over a data-link layer protocol (layer 2) such as Ethernet should suffice to demonstrate that ARP is not naturally and necessarily present over Ethernet, as would be required to find inherency. The Examiner's Answer then continues down the path of attempting to fabricate anticipatory elements in Bennett by alleging: "Both the TCP/IP packet and the ARP request frame are prepared at the host CPU and handed down to the CPD for transmission". While appellants find it difficult to argue with elements that are conjured out of the Ether (no pun intended), suffice it to say that an ARP request would not have a TCP port that corresponds to the TCP connection as recited in claim 1. For at least these additional reasons, appellants believe that the new allegations of purportedly anticipatory elements in the Examiner's Answer are, to quote the Examiner, "misguided".[6]

The Examiner's Answer next provides new phrasing in an attempt to comport with its newly coined interpretations of a "TCP connection" and "to create a TCP connection", but this only serves to point out further inconsistencies with those interpretations. For instance, because the Examiner's Answer points to the same disclosure of Bennett (construction of an ACK packet) as allegedly teaching both "to create a TCP connection" and "when said CPD controls said connection and said message corresponds to said connection", the Examiner's Answer ignores the separate limitation of claim 1 that is found in the "wherein" clause of that claim. Moreover, setting aside the Examiner's Answer's curious new interpretations involving a "TCP connection", the limited processing taught by Bennett for bouncing back an ACK is by Bennett's own

[4] Examiner's Answer, page 4, lines 9-10.
[5] 37 CFR 41.39(b)(2); 37 CFR 41.41(a)(2).
[6] Examiner's Answer, page 9, line 10.

description not the primary processing for that packet,[7] let alone for the other related packets that may be concurrently processed. This falls far short of the limitation in claim 1 that "said CPD controls said connection".

For at least these additional reasons, appellants respectfully assert that Bennett does not anticipate claim 1 or any claim that depends from claim 1.

As noted above, the Examiner's Answer begins on page 3 to provide some interesting new arguments for the rejection of claim 1. On page 7, however, the Examiner's Answer begins what is titled "Response to Arguments", to which appellants reply below.

At the outset of the "Response to Arguments" the Examiner's Answer mischaracterizes appellants' nonenablement argument, which the Examiner's Answer then unsuccessfully attempts to refute. To present this "strawman" argument, the Examiner's Answer states essentially that appellants argue that Bennett's automatic ACK generation "would not be able to carry out TCP's sliding window algorithm", and then tries to knock down that "strawman" by showing that Bennett discloses a "window" field.[8] While it is true that Bennett's automatic ACK generation would cause errors in the TCP sliding window protocol, despite disclosing a "window" field, the errors that would be caused by Bennett's automatic ACK generation are more fundamental than mere errors in the TCP sliding window protocol. As noted in the Appeal Brief,[9] Bennett's automatic ACK generation would destroy the reliability of TCP by communicating that packets were received when in fact they were lost.

The sliding window protocol was discussed in the Appeal Brief because, despite Bennett repeatedly and clearly alleging that its automatic ACK generation would significantly improve system performance,[10] the Examiner alleged that Bennett instead teaches a primitive stop-and-wait procedure that would slow TCP communication to a crawl. Bennett, of course, never teaches that a stop-and-wait procedure must be used in

[7] See, e.g., Appeal Brief, page 12, line 1 – page 13, line 13.
[8] Examiner's Answer, page 7, lines 10-13.
[9] Appeal Brief, page 6, line 1 – page 11, line 13.
[10] See, e.g., the "SUMMARY OF THE INVENTION", column 1, line 66 – column 2, line 18.

order to avoid destroying data integrity. Indeed, there is simply no teaching or suggestion in Bennett of the stop and wait procedure proposed by the Examiner.

Moreover, the stop-and-wait procedure concocted by the Examiner would require all remote nodes, rather than Bennett's proposed local node that automatically ACKs incoming packets, to implement the stop-and-wait procedure in order to communicate reliably with the local node, and there is no hint in Bennett of instructing any of the remote nodes to do this. That is, the Examiner's proposed modification of Bennett would require any remote node transmitting to a local node containing Bennett's automatic ACK generation to somehow know that it must slow TCP transmission to a crawl by sending a single packet at a time, so that Bennett's automatic ACK generation could respond with a single ACK at a time, the receipt of that ACK by the remote node finally allowing the remote node to send the next packet. The Appeal Brief pointed out that the sliding window protocol, which has been part of TCP for decades, was designed to avoid the primitive communication method espoused by the Examiner. That is, the proposed modification of Bennett by the Examiner to slow TCP communication to a crawl is in contrast to the sliding window protocol of TCP, which has from the inception of TCP allowed the transmission of multiple packets by providing a way for those multiple packets to avoid overwhelming the receiver's memory.

Aside from Bennett's automatic ACK generation, there is nothing else novel in Bennett to which one can attribute Bennett's audacious claim of significantly improving performance, so it is clear that Bennett does not teach such a stop and wait procedure. Even if there was such a teaching, there is no teaching or suggestion in Bennett of having a remote node perform such a stop and wait procedure so that it could communicate with a local node having is Bennett's automatic ACK generation.

Moreover, even the "strawman" argument advanced by the Examiner's Answer directly contradicts the Examiner's contention that Bennett teaches a stop-and-wait procedure. As taught by Bennett, "WINDOW is the amount of memory available for receiving data through the connection between FTP server process 55 and FTP client process 85. In this example, WINDOW is the amount of unallocated space in buffer 53 as shown in FIG. 4."[11] There would be no need for a window field for the Examiner's proposed stop-and-wait procedure, because there would always be memory available for the single packet the Examiner proposes sending, receiving and ACKing.[12] Therefore, a window field would not exist in Bennett, if Bennett functioned according to the Examiner's proposed stop-and-wait procedure, because having the window field in the "ACK command" would consume limited resources that would be wasted on the ACK command. Thus, the very field pointed to by the Examiner's Answer to provide support for the conclusion of its "strawman" argument instead contradicts that conclusion.

The Examiner's Answer then states that Bennett is aware of the issue of out of order packets and provides a buffer to temporarily store these types of packets for subsequent reassembly, "which shows that reception of an out of order packet is not acknowledged until all the relevant packets are reassembled and placed in their sequential order".[13] Appellants agree that Bennett teaches a buffer for reassembly of out of order packets, but strongly disagree with the Examiners allegation that this "shows that reception of an out of order packet is not acknowledged until all the relevant packets are reassembled and placed in their sequential order". There is simply no support in Bennett for this latter allegation. Stated differently, Bennett appears to be aware of the issue of out of order packets needing to be reassembled, but unaware that TCP employs a cumulative acknowledgement scheme.

The portions of Bennett cited in the last two lines of page 7 of the Examiner's Answer[14] do not teach otherwise, and particularly do not show "that reception of an out of order packet is not acknowledged until all the relevant packets are reassembled and placed in their sequential order". That is, column 11, line 66 – column 12, line 6 of Bennett states: "In the preferred embodiment of the invention, the network card 2000

---

[11] Bennett, column 17, lines 3-6.
[12] If there were only enough total memory to receive a single segment, that segment would consume the available memory so that the window field would always be zero, and the ACK would tell the node that was sending data that there was no room to receive any more data, halting communication. As long as there is more total memory than this minimal amount, there should always be memory available for receiving the single segment sent according to the Examiner's stop-and-wait procedure.
[13] Examiner's Answer, page 7, line 19 – page 8, line 1.
[14] "col. 11 line 66-col.12 line 6; claims 42 and 53

handles the reassembly of fragmented datagrams, and the buffering of segments to process out of order delivery of segments. Network card 2000 generates checksums for incoming TCP/IP and IP segments, and conditionally sends the segment to processes executed by CPU 10, depending whether a checksum result indicates that the segment is uncorrupted." Nothing in this paragraph supports the Examiner's allegation that "reception of an out of order packet is not acknowledged until all the relevant packets are reassembled and placed in their sequential order". Claims 42 and 53 both recite that "said network processing subsystem further buffers inbound datagrams received out of order." Nothing in these claims support the Examiner's allegation that "reception of an out of order packet is not acknowledged until all the relevant packets are reassembled and placed in their sequential order". Because the cited portions of Bennett provide no support for the Examiner's allegation, the Examiner is reminded of his admonition against "intentional distortions".[15]

The Examiner's Answer then states that "there is nowhere in the entire disclosure...Bennett explicitly teaches to issue an ACK to any noncontiguous TCP packet".[16] The Examiner's Answer thus seems to require Bennett to positively state that it is nonenabled, rather than accept logical conclusions that follow from the known operation of TCP. Appellants disagree with the Examiner's premise that in order to be disqualified as prior art, Bennett would have to explicitly proclaim that it is nonenabled. Because TCP is known to send multiple packets, some of which could be lost or arrive out-of-order, and Bennett teaches significantly enhancing performance by automatically acknowledging packets without regard to their order, Bennett teaches generating ACKs for noncontiguous TCP packets. As noted previously, this would inform the sender that missing packets were received when in fact they were not, and would even create missing packets from out-of-order packets. By analogy, Bennett also does not explicitly propose to destroy a TCP connection, although that would be the result of following Bennett, *assuming arguendo* that Bennett would ever achieve a TCP connection to begin with. The fact that Bennett does not state that it is nonenabled, such as in "the scenarios

[15] Examiner's Answer, page 8, line 9.
[16] Examiner's Answer, page 8, lines 2-3.

illustrated in Fig. 12 and claim 43 (at col.25 lines 32-34)"[17] does not disprove the ubiquitous TCP condition of communicating multiple packets with cumulative acknowledgments,[18] in which case Bennett would destroy the reliability that is essential to TCP.

The Examiner's Answer then alleges, on page 8, lines 5-8: "And, as a matter of fact, even if Bennett chooses to use the much criticized 'stop-and-wait' algorithm. It is still an appropriate TCP protocol because it satisfies the sliding window protocol by setting the window size to one; it may be inefficient, but definitely not non-enabled". Appellants respectfully note, however, that Bennett not only does not "choose" a "stop-and-wait" procedure but instead claims as its "SUMMARY OF THE INVENTION":

> It is an object of the present invention to ***provide an improved method and apparatus for efficiently operating a reliable communication protocol*** in a computer network. In particular, ***the present invention provides an improved method and apparatus for constructing a portion of an acknowledgement signal to be sent upon successful receipt of an incoming signal.***
>
> ***In accordance with the preferred embodiment of the present invention***, a protocol processing subsystem receives data from another computer in a computer network. Upon successful receipt of the data, the protocol calls for construction of an acknowledgement signal to be sent from the receiving computer to the sending computer. ***The protocol processing subsystem of the present invention automatically constructs this acknowledgement signal. In the preferred embodiment, the construction of this acknowledgement signal occurs in hardware which alleviates the need for the protocol processing software to construct the acknowledgement signal. This results in a significant improvement in system performance over systems according to the prior art.*** [19]

Appellants respectfully assert that one of ordinary skill in the art would have been guided by Bennett away from attempting the "stop-and-wait" procedure proposed by the Examiner, because such a slow and primitive procedure would not "provide an improved method and apparatus for efficiently operating a reliable communication protocol", and would not result in "a significant improvement in system performance over systems

[17] Examiner's Answer, page 8, lines 3-4.
[18] Claim 43 of Bennett relates to reveiving (sic) an ACK rather than generating an ACK. Fig. 12 of Bennett at most demonstrates Bennett's ignorance of the TCP condition of communicating multiple packets with cumulative acknowledgments.
[19] Bennett, column 1, line 66 – column 2, line 18, emphasis added.

according to the prior art", as stated by Bennett. Appellants further respectfully assert that one of such skill also would not have considered the Examiner's proposed approach because it would drastically slow TCP communication, and would have somehow required all remote nodes to adopt the stop-and-wait procedure. Moreover, because the automatic ACK generation is both noted by Bennett to be the essence of that invention and promoted by the Examiner as allegedly anticipating the claims at issue, the inoperability caused by that automatic ACK generation removes that reference as nonenabled prior art. Appellants note that the Examiner's Answer again reverts to its "strawman" argument that satisfying the sliding window protocol guarantees enablement, but this argument is as unavailing here as it was above.

The Examiner's Answer then alleges, on page 8, lines 9-13:

> The point is: to assert that a prior art is non-enabled one must find explicit evidence to support such allegation without intentional distortions. Applicant is found to have overly emphasized the phrase "automatically writes an ACK" [see col.12 lines 10-11] without considering the fact ACK commands are "automatically" written into a command list buffer for further consideration. For example, some ACK commands could be subsequently suppressed [see, e.g., claims 8, 11, 17, 24, 26 29 and 33].

Appellants are bemused that the Examiner's Answer admonishes appellants against "intentional distortions" and then alleges, in the sentences immediately following, that Bennett teaches some "facts" that are curiously absent from that reference. To wit, it is misleading in this context to state that Bennett teaches the "fact" that ACK commands are automatically written into a command list buffer for "further consideration". Bennett instead teaches that "ACK commands" in "command list 42" are automatically sent out piggybacked upon related outgoing packets or, if no such outgoing packets are sent within 200 milliseconds at most, upon the expiration of a 200 millisecond timer.[20] There is no "further consideration" as to whether the ACK itself is erroneous, as implied by the Examiner's Answer. Moreover, an out-of-order packet could easily arrive after the expiration of the timer, within which time an erroneous ACK would have automatically been sent, and a lost packet of course would also be masked by the automatic generation of an erroneous ACK.

[20] Bennett, column 19, line 59 – column 20, line 64

In addition, none of claims 8, 11, 17, 24, 26 29 and 33 recite that ACK commands could be "subsequently" suppressed, as alleged in the Examiner's Answer. Because the term "suppress" is curiously absent from the specification of Bennett, it is unclear what these claims teach. Appellants note, however, that the automatic ACK generation proposed by Bennett would only perform simple checksum verification, after which the TCP header would be processed by TCP process 91 on local node 1000,[21] which would generate another ACK for the same packet unless somehow instructed otherwise. Perhaps this duplication of ACKs is what is "suppressed" according to these claims. Although Bennett does not teach how such suppression could occur, casting doubt on the validity of those claims, Bennett may have recognized this duplicate ACK problem, albeit without achieving a solution to that problem. In any event, appellants note that the claims do not teach "subsequent" suppression as an example of "further consideration", as alleged by the Examiner's Answer, and the Examiner is again reminded of his admonition against "intentional distortions".

The Examiner's Answer then alleges, on page 8, line 15 – page 9, line 3:

> Applicant is reminded that at col.12 lines 7-11 Bennett teaches that the condition to automatically send an ACK command to a command list [notice that the command list is stored in a buffer for later consideration] is "reception of a <u>complete</u> IP datagram and a <u>valid</u> checksum result" (underline emphasis added), while at col.16 lines 19-26, Bennett uses "[u]pon <u>successful defragmentation</u> of the datagram and <u>validation of all applicable</u> checksums" (underline emphasis added) as preconditions for sending an ACK. Since out of order packets are resulted in a buffer as depicted at col.11 line 66 – col.12 line 6, it is clear that these successfully received and strictly validated for all applicable checksums packets, for which Bennett's system generates their respective ACK commands by way of either automatically forming at the network card [col.12 lines 7-11] or composing at the host [col.12 lines 11-20], are in fact contiguous, and consequently the acknowledgement itself is cumulative.

Although this paragraph is difficult to understand due to grammatical issues, as best as can be understood the Examiner appears to be arguing essentially that, because Bennett teaches strict validation with checksums, Bennett teaches cumulative acknowledgements. To promote this argument the Examiner appears to be defining

[21] See, e.g., Bennett, column 14, lines 37-43.

cumulative acknowledgements as ACKs that are somehow "contiguous"[22], as opposed to the textbook definition that an ACK verifies to the sender that all data prior to the ACK number has been successfully received.[23] One way to look at this "redefinition" is that this is another "strawman" argument, which in this case redefines a textbook term to fit that which the Examiner can find in the reference. Another way to look at this "redefinition" is in terms of the "intentional distortions" the Examiner warned against. In like vein is the Examiner's statement that the command list is stored in a buffer for "later consideration", as noted above.

This paragraph of the Examiner's Answer also provides some underlined terms that do not change any of the above analysis, but due to their emphasis deserve comment. Appellants respectfully assert that "a complete IP datagram" is simply a complete packet, which can be received out-of-order compared to other packets.[24] A "valid checksum result" simply means that the simple checksum arithmetic for a packet has been verified. The "successful defragmentation of the datagram" refers to the situation in which a datagram, or packet, is received in fragments that are later successfully put back together to form the packet, which again does not preclude that packet from being received out-of-order, or other packets from being lost or received out-of-order. Finally, "validation of all applicable checksums" simply refers to the fact that a checksum can exist for a header as well as an entire packet.

Thus, the above quoted paragraph from the Examiner's Answer is both factually and logically flawed, and at best ignores appellants' arguments and evidence that show that out-of-order or lost packets would be erroneously acknowledged as having been received, provided that checksums of other packets are validated by Bennett's automatic ACK generation.

The Examiner's Answer then alleges, on page 9, lines 4-12:

> As another point of hint, Applicant is further reminded that since Bennett's system is focused on using a network interface card to replace the role of the host CPU in the ACK packet generation, it is a common sense that whatever algorithm works at the host CPU for the job should be implemented in the network card as well. Thus there is no need to disclose

[22] Examiner's Answer, page 9, lines 1-3.
[23] See, e.g. Appeal Brief, page 6, line 27 – page 7, line 5; page 9, lines 16-25.
[24] See, e.g., Bennett, column 6, lines 48-61.

> all the implementation details that a skilled artisan has already known. Bennett's teaching may have been incomplete in that it does not explicitly discuss the issue of lost packets and thereby may cause certain readers to be "misguided", but it is definitely applicable at least to the exemplified scenarios wherein issuing ACK packets and sending subsequent packets are nicely interleaved [see Fig. 12 and claim 43].

Although this argument is sensibly couched in terms of "common sense", appellants respectfully submit that it is the complexity of TCP that leads to Bennett's nonenablement and the Examiner's misunderstandings with the requirements of TCP and the facts of Bennett. Indeed, the automatic ACK generation taught by Bennett and extolled in Bennett's "SUMMARY OF THE INVENTION" directly contradicts the cumulative acknowledgement protocol that is part of TCP. Stated differently, there is no "common sense" solution based on the way the CPU handles ACKs, because the CPU does not automatically generate ACKs without regard to whether the cumulative data represented by the ACK as having been received has in fact been received.

Regarding Fig. 12 and claim 43 of Bennett, as noted above, claim 43 relates to reveiving (*sic*) an ACK rather than generating an ACK, and provides no assurance that the ACK is not erroneous, and Fig. 12 at most demonstrates Bennett's ignorance of the TCP condition of communicating multiple packets with cumulative acknowledgments. Stated differently, a primary attribute of TCP is the guarantee of error free data, and the possibility that a mechanism may not cause errors in one extremely restricted, extremely inefficient situation does not mean that the mechanism is operable when it clearly would fail under normal operating conditions.

Because the Examiner's Answer directs unfounded allegations of "intentional distortions" at appellants while promoting factually challenged interpretations of Bennett, it is clear that the Conferees who signed the Examiner's Answer are biased against appellants. It is unfortunate that the Supervisory Examiner in this case happens to be the same Examiner who earlier allowed Bennett, which would tend to bias him against the argument that Bennett is nonenabled. Indeed, it is because of this kind of bias that Reexaminations are not conducted before the same Examiner who earlier allowed the patent under question. Of course, in a Reexamination scenario a patent is under question due to prior art that was not considered before grant, often found by a defendant in a litigation where the stakes are much higher and the search more thorough, whereas in the present case nothing new is before the Examiner except for the argument that the patent has serious problems which were not identified before grant, which would tend to make the bias much stronger.[25] Needless to say, participation as a Conferee[26] in the Examiner's Answer by that same Supervisory Examiner may not lead to the most unbiased arguments in this case. This bias would likely add to the bias which may naturally be felt by the Primary Examiner when his reasoning is questioned and his primary reference is shown to be nonenabled.

As noted above, the Examiner's Answer on page 3 provides some curious new arguments for the alleged anticipation of claim 1. After arguing that Bennett is enabled under the heading "Response to Arguments", as discussed above, the Examiner's Answer returns to the allegation of anticipation of claim1 by stating, on page 9:

> As to claim 1, Applicant argues that (1) Bennett's CPD (i.e., the network card) does not control the TCP connection; (2) Bennett's CPU does not provide the CPD a media access control (MAC) address; and (3) Bennett's CPU does not create a TCP connection between the local host and the remote host.

Appellants have above replied to these arguments as they were initially presented by the Examiner's Answer in the form of very curious interpretations of some of the terms of claim 1. For thoroughness appellants reiterate and expand those replies below, in response to the following arguments of the Examiner's Answer:

> As to point (1), it is noted that Applicant does not have a clear and precise definition as to what is meant to be a "TCP connection", the term has thus been intuitively construed as a physical connection established for transmission of a TCP packet. That is, between the host CPU and the network card, whoever is granted (or arbitrated) to independently transmit/receive a TCP packet via the established TCP connection must have controlled the TCP connection for at least that particular moment. Since Bennett's network card has the capability of independently sending/receiving an ACK packet (which by itself is a TCP packet), the

[25] Note, however, that nonenablement of Bennett as an anticipating reference exists because the disclosure of Bennett does not enable one of ordinary skill in the art to practice the claims on appeal, as opposed to the issued claims of Bennett.
[26] Examiner's Answer, page 8, lines 3-4.

> network card must have, at least for that particular moment, controlled the TCP connection.[27]

Initially, appellants note that there is no requirement that the specification define claim terms, let alone have "a clear and precise definition of" a claim term, as stated by the Examiner's Answer. The Examiner's Answer exploits this false requirement,[28] however, to state that "the term (TCP connection) has thus been intuitively construed as a physical connection established for transmission of a TCP packet". To create this interpretation, the Examiner's Answer had to ignore the specification,[29] conflate the physical layer (layer 1) and the transport layer (layer 4), ignore other portions of the claim that conflict with his interpretations,[30] and contradict the understanding of those of ordinary skill in the art. For example, the present application states that "Hundreds of TCP, TTCP, SPX or similar CCB connections may be held indefinitely by the INIC, although a least recently used (LRU) algorithm is employed for the case when the INIC cache is full",[31] which contradicts the Examiner's interpretation of a TCP connection as "a physical connection established for transmission of a TCP packet", because such "a physical connection" could not be held in a cache on the INIC.

The Examiner's Answer continues its new arguments by stating:

> As to point (2), it is noted that Bennett's teachings apply to any computer nodes illustrated in Figs. 1 and 10 including, for example, Ethernet nodes 272 and 276 [col.3 lines 45-59] wherein an Ethernet interface [51, Fig.11A] is built into each network card. Frames or packets resulted from higher level activities such as TCP, ICMP and ARP protocol processing {e.g., 90, Fig. 2A, wherein ARP is listed in the transport layer of Ethernet node 276] are prepared at the host CPU except for the portions that Bennett explicitly teaches to be processed at a special hardware in the network card. This ARP data flow assures that at least a source MAC address is included in the Ethernet-bound ARP request frame. Although

[27] Examiner's Answer, page 9, line 18 – page 10, line 4.

[28] Note the similarity of this false requirement with responding to a false premise of a "strawman" argument, and the hypocrisy of first requiring that one look to the specification for a precise definition, and then completely ignoring the specification if it does not state a "precise definition".

[29] See. e.g., page 11, line 12 – page 12, line 3; page 18, lines 6-9; page 21, line 23 – page 22, line 10; page 24, lines 1-11.

[30] See page 2, lines 10-25 above.

[31] Page 36, lines 2-4.

> Bennett does not specifically teach the operation of an ARP, it is submitted that ARP is an inherent function to Bennett's Ethernet nodes such as nodes 272 and 276 and that the Ethernet nodes in Bennett's Figs. 1 and 10 must have provided a MAC address to the CPD in an ARP request frame.[32]

Appellants initially note that FIG. 2A of Bennett represents "remote node 276",[33] as opposed to "local node 1000"[34] of FIG. 2B which contains "TCP logic 93" and "ACK logic 115" on "network card 2000".[35] Similarly, FIG. 11A of Bennett represents "remote node 276",[36] which does not contain "network card 2000".[37] Thus, even *assuming arguendo* the accuracy of what the Examiner's Answer states in this paragraph, the network card 2000 would be located remotely from the hypothetical ARP request, so that a MAC address is not taught to be provided to the "network card 2000", in contrast with claim 1. Moreover, even *assuming arguendo* if a MAC address was taught to be provided to the network card, there is no assurance that the MAC address would correspond to a TCP connection. Indeed, because the Examiner's Answer defines a TCP connection as "a physical connection established for transmission of a TCP packet," the hypothetical ARP request would not have the same MAC address as that occupied by the TCP connection.

Appellants also question the Examiner's Answer statement that "Frames or packets resulted from higher level activities such as TCP, ICMP and ARP protocol processing [e.g., 90, Fig. 2A, wherein ARP is listed in the transport layer of Ethernet node 276] are prepared at the host CPU except for the portions that Bennett explicitly teaches to be processed at a special hardware in the network card." This statement contradicts the Examiner's argument that "when a system uses network card 2000 (of Fig.3) to replace the related processing that is otherwise handled by TCP/IP software, the message related to TCP ACK is totally processed at the CPD".[38] To make the statement that "the message related to TCP ACK is totally processed at the CPD", the Examiner

[32] Examiner's Answer, page 10, lines 5-14.
[33] *Id*, column 4, line 10.
[34] *Id*, column 4, line 16.
[35] *Id*, column 4, lines 60-65.
[36] *Id*, column 12, line 45.
[37] *Id*, column 12, line 52.

had to ignore various portions of Bennett that teach that the message related to TCP ACK is ***not*** totally processed at the CPD.[39] In other words, the Examiner argues both that a transport layer message is "prepared at the host CPU except for the portions that Bennett explicitly teaches to be processed at a special hardware in the network card", and "totally processed at the CPD", depending on which argument suits his needs of the moment.

Appellants also respectfully disagree with the Examiner's allegation that "ARP is an inherent function to Bennett's Ethernet nodes such as nodes 272 and 276 and that the Ethernet nodes in Bennett's Figs. 1 and 10 must have provided a MAC address to the CPD in an ARP request frame." ARP is simply another protocol besides TCP, and the fact that ARP is not a necessary transport level protocol is shown in the very drawings cited by the Examiner, wherein for example, FIG. 2A shows node 276 that contains ARP and FIG. 2B shows local node 1000 that does not contain ARP. Moreover, appellants object to the Examiner's new argument of inherency in the Examiner's Answer, because appellants are not allowed to introduce evidence to refute that argument, although as discussed above the very item cited by the Examiner actually shows that ARP is not naturally and necessarily present in all nodes, as would be required for inherency.

The Examiner's Answer continues its new arguments by stating:

> As to point (3), it is noted that Applicant does not specifically teach what is meant by "create a TCP connection" [note that creating a "connection context" is different from creating a connection], the term is being construed as "schedule for transmitting a TCP packet over a physical connection" or "cause a TCP transmission to happen". And for reasons shown in point (1) above, it is submitted that Bennett's CPU and network card are intermittently scheduled to performing TCP transmissions or occupying a TCP connection), therefore a TCP connection is created for each of the TCP transmission moment. As for the feature of "running protocol processing instructions", it is noted that the passage recited from col.4 lines 22-50 clearly shows an activity engaging a set of protocol processing instructions necessary for preparing a TCP packet (e.g., via processes 91 and 96), and releasing or regaining a TCP connection in a time shared manner [i.e., the time sharing scheme for sharing the TCP connection resource between the CPU and the CPD is a protocol, and the act of such a time sharing scheme is an act of "running protocol processing instructions"].[40]

---

[38] Final Rejection, page

[39] See, e.g., Appeal Brief, page 12, line 1 – page 13, line 13.

[40] Examiner's Answer, page 10, line 15 – page 11, line 3.

Similar to the Examiner's curious new interpretation of "a TCP connection", which is discussed above, the Examiner's Answer ignores basic tenets of claim construction[41] to construe that to "create a TCP connection" means "schedule for transmitting a TCP packet over a physical connection" or "cause a TCP transmission to happen". That is, the Examiner's Answer makes up a definition which contradicts the specification,[42] which contradicts other terms of the claim, which lacks any evidentiary support, and which would not make any sense to a person of ordinary skill in the art. For instance, the Examiner's new definition that "create a TCP connection" means "cause a TCP transmission to happen" would mean that each ACK that was generated by Bennett's "network card 2000" would create a new TCP connection, which therefore could not correspond to a "TCP connection" that was created on the host by transmission of a different packet. Moreover, the Examiner's new definition that to "create a TCP connection" means "cause a TCP transmission to happen" renders superfluous the wherein clause of claim 1, because according to the Examiner's new definition an ACK transmission would always be "processed by said CPD instead of said CPU" and would always have the conditions that "said CPD controls said connection" and "said message corresponds to said connection". The Examiner's other new definition that to "create a TCP connection" means "schedule for transmitting a TCP packet over a physical connection", in addition to suffering from the problems listed above, also makes no grammatical sense, and so would be rejected by lay persons as well as those of ordinary skill in the art. Indeed, the only basis for the Examiner's curious definitions appears to be that he believes that he can find elements in Bennett that anticipate these definitions.

Regarding the recitation of "running protocol processing instructions", the Examiner's Answer cites "processes 91 and 96" in Bennett, which are not part of "network card 2000". This contradicts the Examiner's other definitions, which require the "TCP connection" to be created on Bennett's "network card 2000" by automatically

[41] See, e.g., *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005).
[42] See, e.g., page 11, lines 12-18; page 15, lines 3-7; page 18, lines 6-12; page 20, lines 11-15; page 21, line 23 – page 22, line 2; page 22, lines 8-10; page 24, lines 1-4; page 24, lines 9-11; page 31, lines 5-7; page 31, line 17 – page 32, line 5; page 35, line 19 – page 36, line 4.

sending ACKs from the card, which in the Examiner's argument equates to controlling the TCP connection.[43] Otherwise the ACKs would create different TCP connections that would not correspond to the "TCP connection" and the card could not control the "TCP connection". But claim 1 recites "a central processing unit (CPU) running protocol processing instructions in the local host to create a TCP connection", so that at least one of the "TCP connections" alleged by the Examiner must be incorrect.

The Examiner's Answer continues its litany of unreasonable interpretations by equating protocol processing instructions with "releasing or regaining a TCP connection in a time shared manner [i.e., the time sharing scheme for sharing the TCP connection resource between the CPU and the CPD is a protocol, and the act of such a time sharing scheme is an act of 'running protocol processing instructions']". This interpretation contradicts the Examiner's other statements regarding "running protocol processing instructions". This also contradicts basic tenets of claim construction by contradicting the specification and the understanding of those of ordinary skill in the art. In short, it is clear that the Examiner's litany of unreasonable interpretations, like his "strawman" enablement argument, is provided so that the Examiner can attempt to achieve the conclusions he desires. Like his "strawman" arguments, however, the Examiner's litany of unreasonable interpretations fail to provide the answer he alleges, partly by contradicting each other even if, *assuming arguendo*, they were to be adopted.

Regarding claims 17, 30 and 42, the Examiner's Answer states that appellants make three points, and then responds to point (1) while referring the reader to other of his arguments for points (2) and (3),[44] to which appellants have already replied. Appellants will therefore respond only to the Examiner's argument regarding point (1), which is found on page 11, line 21 – page 13, line 10 of the Examiner's Answer.

Nothing in the Examiner's argument regarding point (1) shows that Bennett teaches "said CPD receiving control of said connection from said CPU", as recited in claim 30. Moreover, appellants respectfully assert that the Examiner's Answer is

---

[43] See Examiner's Answer, page 4, lines 17-21.
[44] Examiner's Answer, page 11, line 4 – page 13, line 14

misleading when it states "As to point (1): Applicant is reminded that at col. 16 lines 19-35, which Applicant's argument is based on". Appellants did not provide a citation regarding "point (1)" because the Examiner did not even allege that "said CPD receiving control of said connection from said CPU", as recited in claim 30. Again the Examiner's Answer has misstated the arguments, and again failed to find support for even its misstated arguments.

Regarding the recitation of "said CPD classifying a second network packet as corresponding to said connection", the Examiner's Answer again misstates the tenets of claim construction by stating: "Note that due to the lack of specific definition in Applicant's specification, the word 'classifying' does not carry any special meaning other than 'identifying' or 'specifying'". The Examiner's interpretation in this regard also contradicts the specification.[45] Moreover, the explanation in the Examiner's Answer that "a message picked up by the CPD and processed by said CPD instead of said CPU (claim 1) is mapped to the second network packet that is identified by the CPD and processed independently therein (claim 30)" makes no sense. Furthermore, this "second network packet" would, according to the Examiner's definition of creating a TCP connection", create yet another "TCP connection" that would not correspond to the one or two other "TCP connections" (depending upon which of the Examiner's contradictions one ignores) that would exist according to the Examiner's interpretation of Bennett and claim 1. Once again, the Examiner has misstated the law to provide support for his interpretation, which fails to anticipate claim 30, even if *assuming arguendo* it is adopted, due to other contradictions with his curious definitions.

Appellants appreciate the Examiner's withdrawal of the rejection of claims 28 and 41, as noted on page 13, lines 15-16 of the Examiner's Answer.

Conclusion

As discussed above, the Examiner's Answer repeatedly misstates the facts, the law and appellants' arguments in an attempt to conclude that the pending claims are anticipated by Bennett. Even these multiple misstatements, however, do not support the

[45] See, e.g., page 29, lines 7-10.

anticipation alleged by the Examiner's Answer, in part because the strained interpretations derived from some of the misstatements contradict each other. Indeed, the litany of misstatements only emphasizes the weakness of the Examiner's arguments. Moreover, correction of the misstatements further compels the conclusion that all the pending claims are allowable over Bennett.

Respectfully submitted,

[illegible]

Mark Lauer
Reg. No. 36,578
6601 Koll Center Parkway
Suite 245
Pleasanton, CA 94566
Tel: (925) 484-9295
Fax: (925) 484-9291

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with sufficient postage in the US Postal Service as first class mail in an envelope addressed to: MS Appeal Brief, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on December 5, 2006.

Date: 12-5-06

[illegible]

Mark Lauer